CLAIMS

1. An object model document for persisting an object model therein, the document comprising a compiled executable file having:

an image source from which the persisted object model is instantiated in a memory of a computer;

a security source from which a security agent is instantiated in the memory of the computer; the security agent for controlling access to the object model as instantiated in the memory of the computer; and

a loader for being instantiated in the memory of the computer upon a command from a commander to execute the executable file to instantiate the persisted object model, the loader for instantiating the object model in the memory from the image source, instantiating the security agent in the memory from the security source, and returning to the commander a first reference to the instantiated security agent, whereby the commander in employing the first reference accesses the security agent rather than the instantiated object model.

2. The document of claim 1 wherein the executable file is compiled by a compiler from a C-type programming language object model document.

3. The document of claim 1 wherein the loader upon instantiating the security agent provides same with a second reference to the instantiated object model, whereby the commander does not have the second reference and therefore cannot directly access the object model or command same to act.

4. The document of claim 1 wherein the instantiated security agent passes on each command from the commander to the object model unless

such security agent deems such command to be of a type that should not be so passed on.

5. The document of claim 4 wherein the security agent does not pass on to the object model a type of command that would expose the object model in a non-obfuscated form.

6. The document of claim 4 wherein the security agent does not pass on to the object model a type of command that would expose the object model with a level of granularity finer than a pre-defined maximum.

7. The document of claim 6 wherein the security agent passes on to the object model a substitute command that exposes the object model with a level of granularity coarser than the pre-defined maximum.

8. The document of claim 1 wherein the loader instantiates the security agent separately from the object model.

9. The document of claim 1 wherein the loader instantiates the security agent as part of the object model.

10. A method for loading a persisted object model from an object model document comprising a compiled executable file having an image source, a security source, and a loader, the method comprising:

instantiating the loader in a memory of a computer upon a command from a commander to execute the executable file to instantiate the persisted object model;

the loader instantiating the object model in the memory from the image source;

the loader instantiating a security agent in the memory from the security source, the security agent for controlling access to the object model as instantiated in the memory of the computer; and

the loader returning to the commander a first reference to the instantiated security agent, whereby the commander in employing the first reference accesses the security agent rather than the instantiated object model.

11. The method of claim 10 further comprising the loader upon instantiating the security agent providing same with a second reference to the instantiated object model, whereby the commander does not have the second reference and therefore cannot directly access the object model or command same to act.

12. The method of claim 10 further comprising the instantiated security agent passing on each command from the commander to the object model unless such security agent deems such command to be of a type that should not be so passed on.

13. The method of claim 12 comprising the security agent not passing on to the object model a type of command that would expose the object model in a non-obfuscated form.

14. The method of claim 12 comprising the security agent not passing on to the object model a type of command that would expose the object model with a level of granularity finer than a pre-defined maximum.

15. The method of claim 14 comprising the security agent passing on to the object model a substitute command that exposes the object model with a level of granularity coarser than the pre-defined maximum.

16. The method of claim 10 comprising the loader instantiating the security agent separately from the object model.

17. The method of claim 10 comprising the loader instantiating the security agent as part of the object model.

18. A computer-readable medium having stored thereon an object model document for persisting an object model therein, the document comprising a compiled executable file having:

an image source from which the persisted object model is instantiated in a memory of a computer;

a security source from which a security agent is instantiated in the memory of the computer; the security agent for controlling access to the object model as instantiated in the memory of the computer; and

a loader for being instantiated in the memory of the computer upon a command from a commander to execute the executable file to instantiate the persisted object model, the loader for instantiating the object model in the memory from the image source, instantiating the security agent in the memory from the security source, and returning to the commander a first reference to the instantiated security agent, whereby the commander in employing the first reference accesses the security agent rather than the instantiated object model.

19. The medium of claim 18 wherein the executable file is compiled by a compiler from a C-type programming language object model document.

20. The medium of claim 18 wherein the loader upon instantiating the security agent provides same with a second reference to the instantiated object model, whereby the commander does not have the second reference and therefore cannot directly access the object model or command same to act.

21. The medium of claim 18 wherein the instantiated security agent passes on each command from the commander to the object model unless such security agent deems such command to be of a type that should not be so passed on.

22. The medium of claim 21 wherein the security agent does not pass on to the object model a type of command that would expose the object model in a non-obfuscated form.

23. The medium of claim 21 wherein the security agent does not pass on to the object model a type of command that would expose the object model with a level of granularity finer than a pre-defined maximum.

24. The medium of claim 23 wherein the security agent passes on to the object model a substitute command that exposes the object model with a level of granularity coarser than the pre-defined maximum.

25. The medium of claim 18 wherein the loader instantiates the security agent separately from the object model.

26. The medium of claim 18 wherein the loader instantiates the security agent as part of the object model.

27. A method for processing a command from a commander to an object model instantiated in a memory of a computer, the commander issuing the command by way of a first reference to a security agent instantiated in the memory of the computer, the security agent for controlling access to the object model as instantiated in the memory of the computer, the method comprising:

the security agent receiving the command from the commander;

the security agent reviewing the command according to pre-defined rules therein to determine whether the object model should in fact receive the command; and

if so, the security agent forwarding the command to the object model and the object model receiving the command and executing same.

28. The method of claim 27 wherein if the security agent determines that the object model should not in fact receive the command, the security agent does not forward the command to the object model.

29. The method of claim 28 wherein if the security agent determines that the object model should not in fact receive the command, the security agent responds to the commander with a message that the command cannot be issued to the object model.

30. The method of claim 27 comprising the security agent forwarding the command to the object model by way of a second reference thereto.

31. The method of claim 27 comprising the security agent not forwarding to the object model a type of command that would expose the object model in a non-obfuscated form.

32. The method of claim 27 comprising the security agent not forwarding to the object model a type of command that would expose the object model with a level of granularity finer than a pre-defined maximum.

33. The method of claim 32 comprising the security agent forwarding to the object model a substitute command that exposes the object model with a level of granularity coarser than the pre-defined maximum.